

# Intégration graphique

Quelques principes qui nous permettent de réaliser une intégration graphique assez facilement.

## Description rapide du fonctionnement de l'affichage Publik

Le système Publik utilise le framework Django, notamment son système de gabarit (templates) :

<https://docs.djangoproject.com/fr/1.9/ref/templates/>

Une instance de Publik définit un thème de base, composé :

- d'un gabarit de base (theme.html), c'est-à-dire une structure HTML de base qui comporte quelques éléments de placement des objets à afficher :
  - zone de connexion, en général un cartouche avec des liens : se connecter, se déconnecter, accès au profil, etc.
  - zone de contenu
- à partir de ce gabarit de base, on définit des gabarits de page, qui précisent la zone de contenu. On a souvent :
  - un gabarit "une seule grande colonne"
  - un gabarit "deux colonnes" (une large + une barre latérale)

Au niveau des portails (portail usager, portail agent), il s'agit de systèmes CMS où chaque page indique le gabarit qu'elle suit. Ensuite, dans chaque zone du gabarit choisi (chaque colonne, souvent) on choisit les cellules de contenu qui seront affichées : un titre, du texte, la liste des démarches, les brouillons de l'utilisateur, les livres empruntés à la médiathèque, afficheur de flux RSS, etc.

Au niveau des applications de Publik (les briques de la solution : gestion des formulaires, porte-document, fournisseur d'identité, etc.) on indique un gabarit à suivre pour chacune. Chaque application affiche alors son contenu dans la zone de contenu du gabarit.

Suivant ces principes, l'ensemble de l'instance Publik suit les mêmes règles de présentation et on a une homogénéité graphique générale de l'instance.

## Création du gabarit de base

Le plus souvent on dispose d'un site existant dont on doit copier le design, c'est-à-dire :

- les feuilles de styles CSS, accompagnées d'un ensemble d'éléments statiques (fontes, icônes, etc.)
- la structure HTML d'une page type

## Gabarit de base

Pour cela, sur le site existant, on détermine une page qui sera le modèle à suivre et on crée un gabarit de base (theme.html) à partir d'elle. Souvent, ce gabarit de base est créée une fois pour toutes par Entr'ouvert.

Mais il est également possible d'avoir un script qui récupère la page du site existant et la modifie automatiquement, avec quelques règles de chercher/remplacer. Ce script peut alors être lancé selon une certaine fréquence (en général 1 heure) afin que des modifications faites sur le site existant se retrouvent sur Publik. Pour l'anecdote, citons le cas d'une ville qui met à jour la météo sur son site en HTML plusieurs fois par jour : au lieu de dupliquer le système complet de téléchargement de la météo, on duplique directement la base avec le contenu HTML correspondant, 4 fois par jour.

Recommandations concernant le site existant (source) :

- idéalement, les règles de rechercher/remplacer pour créer automatiquement le gabarit s'appuient sur des phrases simples à repérer dans la page de base du site existant. Par exemple un code `***CONTENT***` sera facilement remplacé par un bloc de gabarit Django.
- les deux zones à clairement identifier de cette façon sont la zone de contenu (`***CONTENT***`) et la zone de connexion (`***LOGIN***`).

## Feuilles de style

Les feuilles de style et les éléments statiques liés (fontes, images) proposés par le site existant sont utilisés directement. On évite de les copier sur le site Publik, d'abord parce que ce n'est pas toujours facile à faire (il s'agit souvent de beaucoup d'éléments liés entre eux présents un peu partout sur le site), mais aussi pour permettre à Publik de s'adapter automatiquement à une modification du style du site existant, sans délai.

Cependant la feuille de style du site existant ne couvrira pas tous les besoins en style de Publik. Pour compléter, on ajoute dans le gabarit de base une CSS dédiée (publik.css). Elle est programmée avec Sass (<http://sass-lang.com/>).

Contraintes sur le site existant (source) : les éléments de style partagés doivent être fournis en HTTPS. Les polices elles doivent être servies avec une entête Access-Control-Allow-Origin.

## **Règle de bonne conduite**

Bien entendu, l'ajout d'un grand nombre de script de mise à jour automatique rend le système fragile.

C'est pourquoi, la règle de bonne conduite reste d'éviter tout ces automatismes, et de parvenir à construire un gabarit de base statique une fois pour toutes au début du projet. En évitant la présence de «gadgets» (tels la météo, les actualités, menus déroulants profonds, etc) on gagne en simplicité, en efficacité et en sécurité.